EE454 Hardware project 3

Project 3 Report

Title: PD and Bang Bang Line Tracking

25 April 2008

David Houngninou & Teresa Stocker

Abstract

This project consists of designing a control system for a robot to track a black line made of electrical tape. The hardware portion uses the sensors mounted on the robot, the Analog to Digital converter, a LCD display, and features controlling the motors on the wheels. The software implemented includes two control algorithms, a Bang Bang Controller and a PD Controller, both which can be selected by using the input bit at P1.5. Additional software includes the Display module as well as an Analog module, to be explained in further detail in the software portion of the report.

Main Components:

(1) HD44780U HITACHI Dot Matrix Liquid Crystal Display

(1) EE 494 Robot with an embedded 8-bit flash microcontroller ATMEL AT89C51RD2

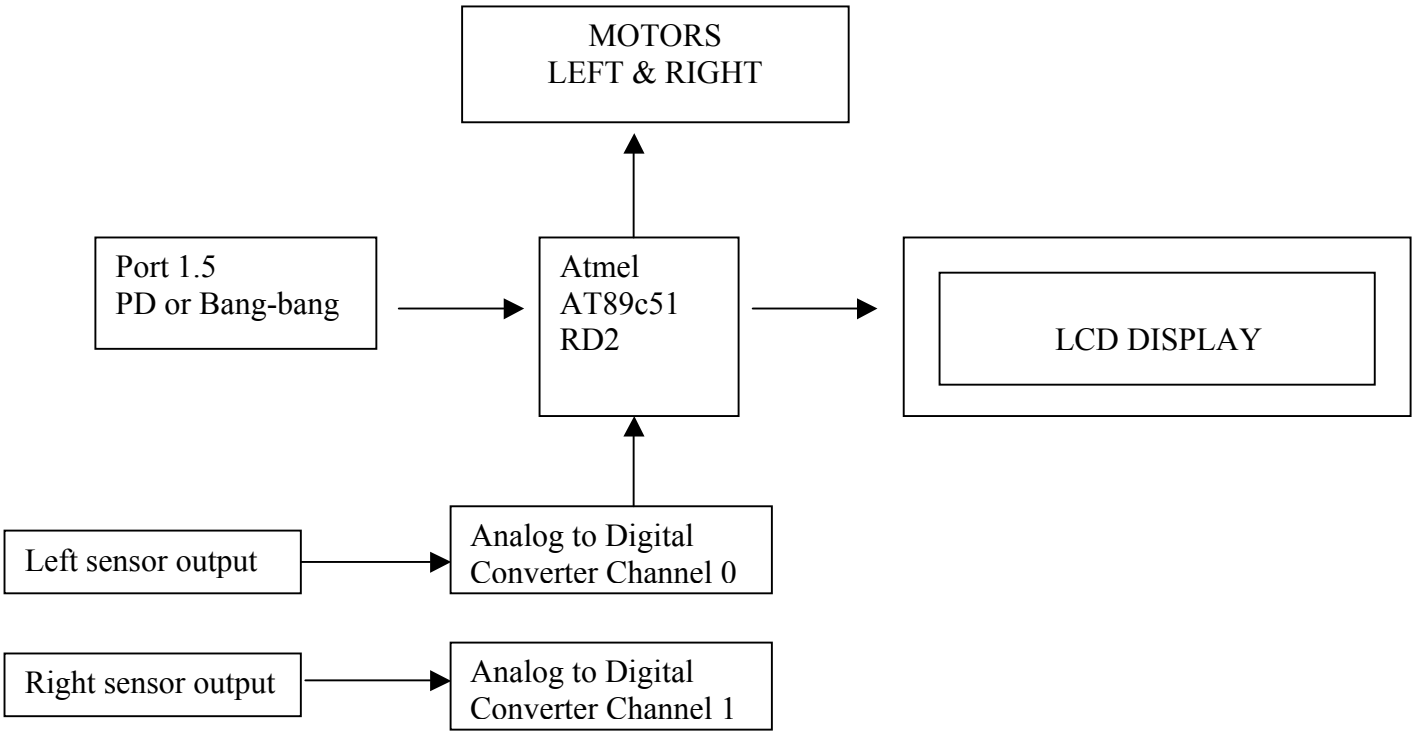


Figure 1: Connection Diagram (Overview)

Description

The program checks on the inputs on pin P1.5 and determines if the user wants to user the PD controller or the Bang Bang. Running in a loop, the code continuously checks the outputs of the left and right sensors at the bottom of the robot to verify the position compared with the electrical tape on the floor.

For the Bang Bang algorithm, if the sensors output indicates that the robot is going off the track, the motors speed are adjusted so that the position gets restored.

For the PD algorithm, the robot is controlled in a proportional mode. The difference in the output between the two sensors is used to generate the amount of control that is steering the robot back to the line and the derivative of the position retards the control that is steering the robot back to the line.

Source Code Documentation:

The software makes use of 5 modules:

Initialize routine design

In the Initialize module, the clock and interrupts are setup. This sets the necessary bits for interrupts if needed, clocks, and timing. Also in this routine, the initialize routine initializes the display screen. There is also a delay routine featured inside the Initialize module that will simply cause a delay. The last feature in this routine is the Motor function. This function will overwrite the speed in CCAP1H and CCAP0H registers if the speed is too high (over 96) or too negative (under -96).

Display routine design

The Display module contains four different functions: Display_Number, Display_Text, LcDelay, and LcDisplay. The latter three functions were taken directly from a sample project and were not changed a large amount. The function that needed altered was Display_Number. This function was initially set to print out 3 digits. In order to display more than three digits, the long int J had to be set larger and re-assigned the value before it entered into the for loop. To compute a digit, the J had to be multiplied by 10 and then divided by itself and converted to hex. J is then divided by 10 and set through the loop to compute the next digit.

Analog routine design

The Analog module communicates with the Analog to Digital port used for the light sensors. It returns the data from the requested channel number after converting it from analog to digital.

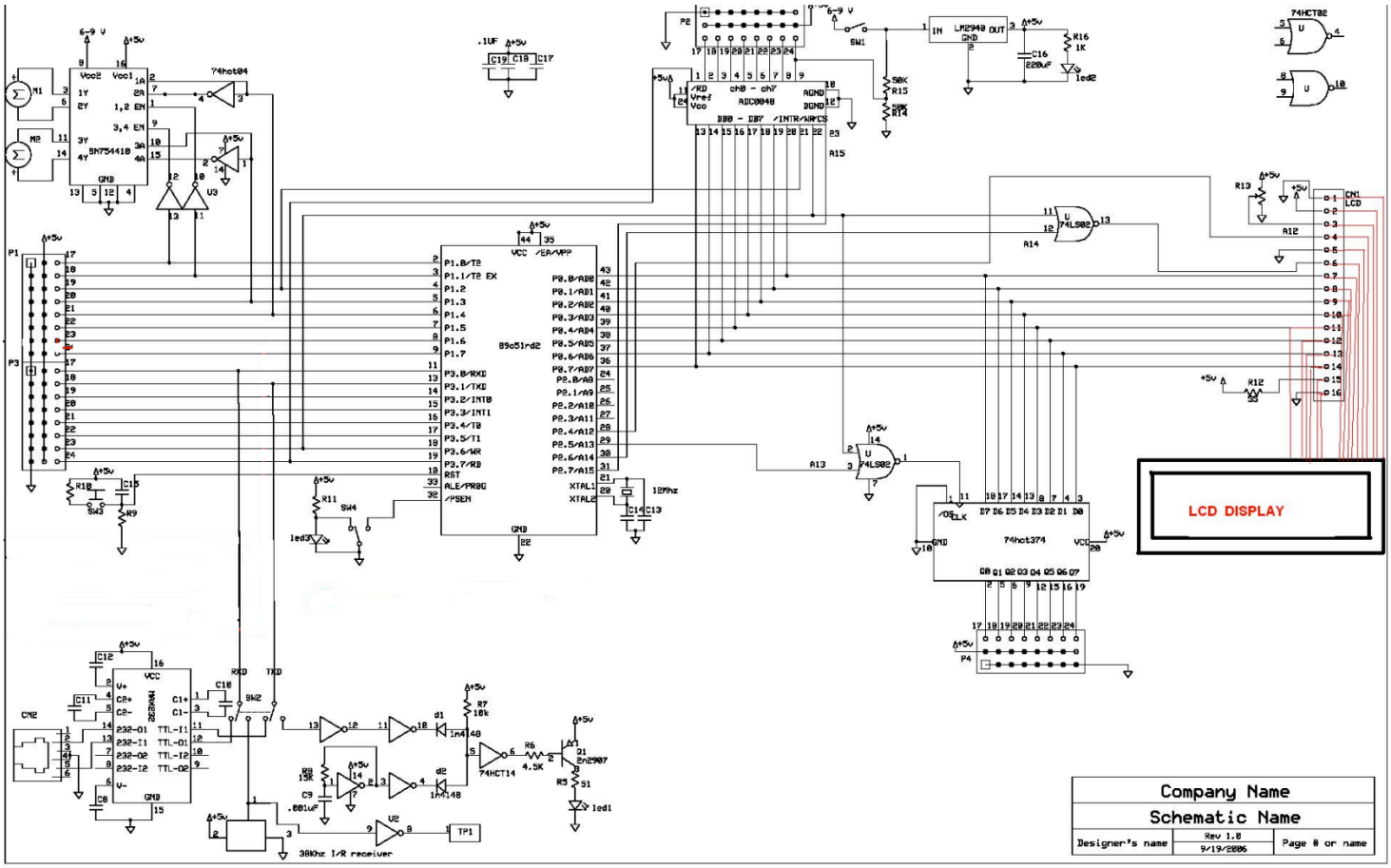
Bang Bang design

In this routine, we start by reading the sensors. Then the Motor function is called to set the wheels turning. This particular function is called with the left motor having a slightly larger number than the right motor because the motors function at different speeds. It continuously checks the values of the sensors to see if the robot is still on the line. If the robot deviates from the line, it will re-adjust the left and right motor speed to get the robot back on the line. Again, the values in the Motor function vary because of the difference in the two motors used.

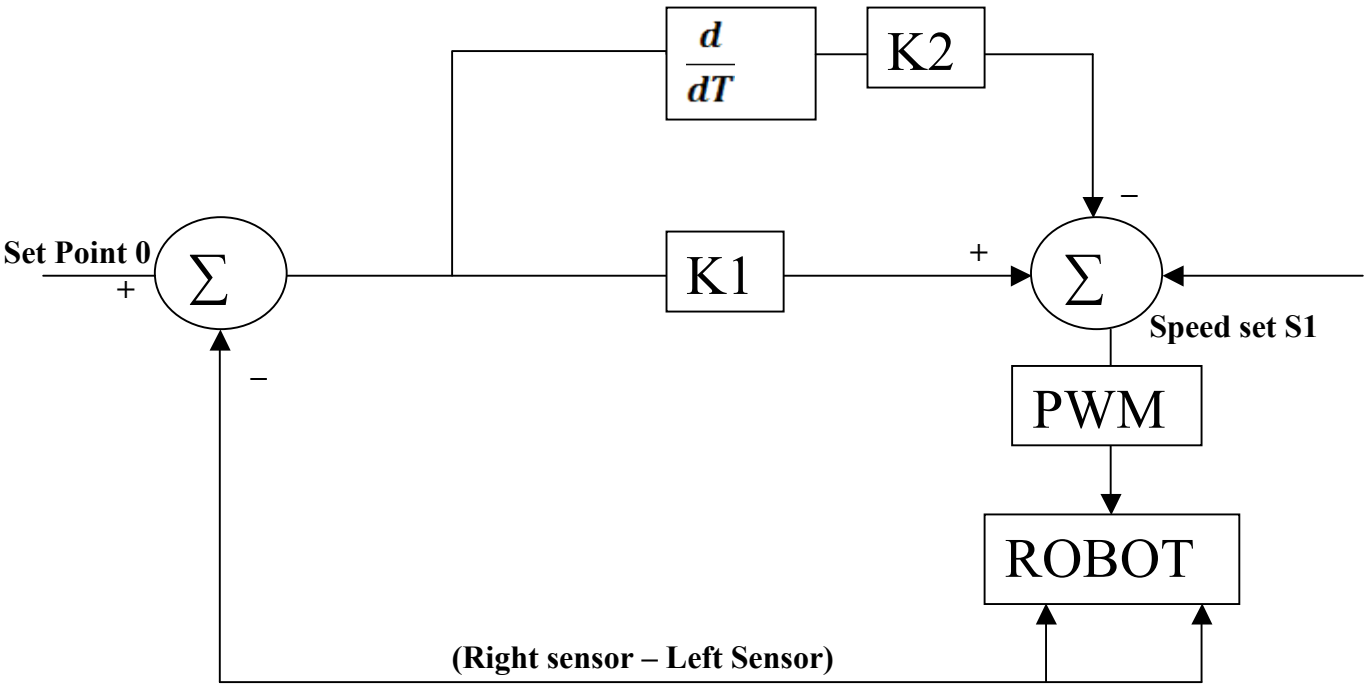
PD Controller design

In this routine, we start again by reading the sensors. We then calculate the position error between the two sensors. The derivative error is calculated by recording the value of the last error and subtracting the difference from the current position error. In order to get a derivative error, the routine is set up with a wait counter that will wait for approximately 10 ms before re-entering the routine. The speed is then calculated from the position error and the derivative error using set constants and then used in the Motor function to set the speed of the left and right motors accordingly. This design produces a much more continuous stream of movement as compared to the bang bang controller design.

Hardware design:



Schematic



Controller Diagram

Extra features

The PD or Bang-Bang algorithm can be selected by using a switch on pin P1.5

When P1.5 is high, the Bang-Bang algorithm runs. When it is low, the PD algorithm runs.

Conclusion

The device has been tested successfully for both the Bang-Bang and PD algorithms and operates at a considerable speed.